# Image Segmentation Algorithms

Aashir Farooq

## 1 Introduction

An image from a camera captures a lot of information about the objects in view. However, we are often interested in specific details about a particular object. A key application of this is in **Medical Imaging** for detecting cancerous tumors. This involves capturing an image of the suspected part of the body and then segmenting and isolating the region of interest. This is where image segmentation becomes crucial. By applying different algorithms, we can extract important information from the image and diagnose the patient based on the findings.

Different algorithms have their specific use cases, and none is perfect for every task. The choice of algorithm depends on the needs of the problem. Some commonly used algorithms include:

### 1.1 Threshold Segmentation

Threshold segmentation separates the background and the target object using a gray-scale threshold value. This can be either a **local threshold** (i.e., multiple segmentations of target and background) or a **global threshold** (i.e., a singular decomposition of target and background). It is computationally simple but lacks accuracy for more complex tasks.

### 1.2 Edge Detection Segmentation

Edge detection segmentation is based on the discontinuity of local color and brightness around the edges of an object. A popular method is the **Laplacian operator**, which is a second-order differential equation. The Laplacian operator separates out the image along its individual axes, making it rotationally invariant. However, this method is computationally lightweight but sensitive to noise. Since it is a second-order differential operator, it often requires noise-free images or the application of a low-pass filter to suppress noise.

The Laplacian equation used in edge detection is:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Where:

- $I$ is the intensity of the image at a given point.
- $\frac{\partial^2 I}{\partial x^2}$ and $\frac{\partial^2 I}{\partial y^2}$ are the second derivatives of the intensity along the x and y directions, respectively.

### 1.3 Clustering-Based Segmentation

Clustering-based segmentation separates pixels based on their proximity or similarity, grouping the image into different clusters. One popular method is **K-means clustering**, which groups pixels based on their distance from a cluster mean. The process involves Initializing cluster centers and calculating the distance between sample pixels and cluster centers, the sample pixel is then assigned to the nearest cluster, the process is repeated until every pixel is assigned to a cluster.

While K-means clustering can be effective, it is computationally expensive and does not account for the spatial connectedness of pixels, making it sensitive to initial conditions.

### 1.4 Computational Neural Networks

By far the most accurate and all-around method to perform image segmentation is through Convolutional Neural Networks (CNNs). More specifically, the U-Net algorithm involves a network of encoders and decoders, which address the "What" and "Where" of the target.

First, the encoder side of the algorithm extracts features of the image using convolution. Then, the spatial dimensions of the extracted data are down-sampled using a $2 \times 2$ Max Pooling Layer. Finally, the ReLU function, which is non-linear, is applied to map the complex image data. This helps identify the target by answering the "What."

Next, the decoder up-scales the samples and uses a Skip Connection to preserve fine details, addressing the "Where" part of the question. A Skip Connection passes feature maps from the encoder to the decoder. For the final output layer, a $1 \times 1$ convolution is applied to map the feature maps onto the original image.

The main benefit of this technique is that it supports both binary and multi-class segmentation without extra effort. It is also considered the best method for image segmentation.

## 2 The best Algorithm

There is no best algorithm for Image Segmentation, each method has its own upsides and applications, if one has to pick, a U-NET algorithm seems to the best fit because :

1. It it extremely flexible can be used for extremely wide range of applications such as Medical Imaging, Satellite Imagery, Facial Identification, Object detection etc, it also has the ability to do Multi-class segmentation without any extra computational cost.
2. Has a better overall accuracy than other methods like Region Based segmentation.
3. It preserves fine details very well and unlike Edge detection algorithms, the input image doesn't need to be noise free.
4. Since it is based on CNN, you can use Pre-trained models to save time, the model does not need to be trained repeatedly.